A Conceptual Framework to Study Socio-Technical Security

Ana Ferreira[1,2], Jean-Louis Huynen[1,2]
Vincent Koenig[1,2], and Gabriele Lenzini[2⋆]

[1] Institute of Cognitive Science and Assessment - Univ. of Luxembourg
[2] Interdisciplinary Centre for Security Reliability and Trust - Univ. of Luxembourg

Abstract. We propose an operational framework for a social, technical and contextual analysis of security. The framework provides guidelines about how to model a system as a layered set of interacting elements, and proposes two methodologies to analyse technical and social vulnerabilities. We show how to apply the framework in a use case scenario.

Keywords: socio-technical framework, security analysis

1 Introduction

Systems that are secure even when used by humans –a property that we call *effective security*– are hard to make. A system can embed technical mechanisms that make it technically secure, such as encryption protocols, but those mechanisms can fail if users bypass or misuse them. Such failures are common since humans do not perceive security as a primary goal [1] and do not properly assess risks when using information communication technology [2, 3]. There is more: computer system designers, with a few exceptions [4], are not accustomed to count human cognitive and behavioural traits as risk factors in the security requirements. Thus, even systems that have been validated as technically secure, may still be insecure against non-technical attacks (e.g., social engineering) remaining oblivious of socio-technical vulnerabilities.

How can we achieve a better effective security? There is no once-and-for-all solution. Effective security is a complex quality to achieve. It is inherently socio-technical (it depends on how human and technical aspects integrate) and it may be context and culture (incl. education) dependent [5, 6]. For example, in hospitals, access control solutions cannot be effective unless designed to fit the nomadic, interrupted, and cooperative nature of the medical work [7]. But, the same access control solutions would be judged differently in a context such as a bank, where employees work mostly alone and where security requirements must consider, for example, threats coming from hackers (e.g., see [8]).

To make a system effectively secure in different scenarios, it likely requires diverse strategies and solutions. However, it is possible to refer to a common

⋆ This research is supported by FNR Luxembourg, project I2R-APS-PFN-11STAS.

framework of analysis. Such a framework should help computer security designers and social scientists to collaborate by providing an operational guideline for an interdisciplinary approach in studying a system's security, as well as tools and methodologies for questioning security at both the technical and the social layers.

Contribution. This paper proposes and describes such framework. STEAL (*Socio-TEchnical Attack AnaLysis*) appears from the need to have a common systematic framework matured from previous experiences the authors had in modelling and analysing socio-technical security [9, 10].

2 Related Work

Zhu *et al.* [11] study how an attacker manages to influence the human to take the wrong decision and acquire his private information. They simulate a scenario where an attacker plays successfully the norm of reciprocity (mutual messages exchange with the user) with the victims who are shopping online with mobile devices. However, this study is incipient and does not provide a systematic way to test and mitigate this or other similar norms. STEAL could model the norm of reciprocity scenario with an overview of all the interactions and maybe provide defences that could be applied in different parts of the system, and not only within the human-computer interface dialogue.

Cranor *et al.* [12, 13] propose a framework to understand how security failures happen when users misbehave because of flawed human-computer communications. This framework is a sequence of generic steps the designer follows to identify potential failure points for each technical function of the system, where the user participates. The designer needs to mitigate those failures, either by eliminating user's intervention altogether if possible, or improving user's interaction. However, there is no specific model/methodology to reproduce both the sequential or the mitigation process and to enable/operationalize scientific-experimental research. Moreover, Cranor's research assumes to know exactly how a technical function will be used by a human and tries to improve it before its usage. So humans are bound by the technology and how a function can be performed, but this may not always be true. The next two works also assume this. Conti et al. [14] research on visualization systems that typically include the human in the decision-making loop and present a visual taxonomy to identify attacks. Falk et al. [15] examine the prevalence of user-visible security design flaws in high security requirements' financial websites, and present a methodology to testing these issues: selecting the most common five security user-visible flaws of website design and identify them in a set of websites. All the above works study the interactions between the user and the computer interface, mostly clarifying usability questions, and not so much enquire about security in all systems' functions and interactions.

Our framework, instead, provides for the design and analysis of socio-technical attacks to the system's functions, humans, context and all its interactions. An attack may exploit bad communications' design but may also ignore technical

functions altogether and focus on the context or the human to perform a successful attack. Moreover, although much research on security usability has been done, these studies are also mostly technology driven.

Other works justify the importance of contextual factors in systems' security in both ATM [16] and hospital authentication solutions [7]. STEAL also integrates context and its interactions in the security analysis.

Regarding social engineering, Janczewski *et al.* [17] review social engineering incidents to give a schematic representation of vulnerabilities usually exploited by social engineers and the attack methodology that better succeeds. Dalpiaz [18] has developed a Socio-Technical Security modelling language which specifies the security and trustworthiness requirements for cross-organizational systems.

Worton *et al.* [19] apply a socio-technical framework to two terrorism scenarios. It groups generic factors like people, goals, technology, culture, buildings, and its characteristics. It is not possible to have a clear overview of how the groups interact and how these interactions could, for instance, generate new threats. Pavkovic and Perkov [20] present SET (for Social Engineering Toolkit), a set of tools to perform advanced attacks against the human element. STEAL could be used to analyse these attacks in more detail.

In summary, we have not found studies that tackle the specific challenge proposed in this paper: to describe a framework providing a common systematic process to analyse the security of socio-technical and contextual factors together with all its interactions. To fulfil this gap, this paper proposes such framework and gives recommendations on how to apply it.

3 A socio-technical security conceptual framework

By a socio-technical security conceptual framework, we mean an operational guideline for a systematic approach in modelling and analysing a system's security in its technical and social perspectives. Past research in security validation shows that important elements of such a framework are (I) a *reference model* and (II) a set of procedural *methodologies*. (I) is to describe, at a suitable level of abstraction, the elements of the system that we intend to analyse. (II) is to have tools for a technical and a social experimental analysis of security.

STEAL, our framework, includes them both (see Fig. 1). Its reference model (see Sec. 3.1) suggests a system as composed by interacting elements/actors (human, interfaces, processes, and context). Its set of methodologies (see Sec. 3.2) includes security validation procedures coming from the formal analysis of security protocols and from the applied cognitive sciences and usability research.

3.1 STEAL: reference model

It is a variant of the Bella *et al.*'s [21] concertina model (Fig. 1, upper part). A socio-technical system is abstractly seen as layered, each layer made of communicating/interacting elements. There is at least a human persona, say Alice

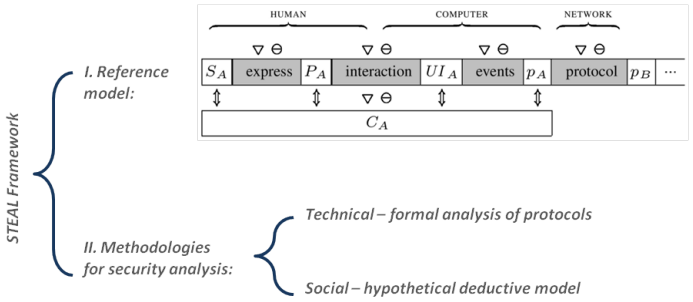

Fig. 1. STEAL Conceptual Framework.

(P_A), and the technology she is using. This is further composed by at least a human interface (UI_A) and some software processes (p_A). Processes can, through a network, communicate with other processes (p_B), behind which may stay one or more humans, say Bob (P_B), who are in turn interfaced to human interfaces (UI_B). Layers can be folded, with the effect that not all elements need to be necessarily in place. Representing our system in this way helps the analyst to select the key components for analysis, and to distinguish between the technical, the human components and the context.

STEAL extends this model by adding the context (C_A), and attack and defence models. Context is the physical or social environment where the interactions for '*Alice*' take place. C_A influences how A's *self* (S_A, in Fig. 1) expresses into P_A's, the way P_A interacts with the interface, and the software, which can be context-dependent. C_B does the same on B's side, not shown in the figure.

This simple reference model fits many scenarios. For example, in a ATM machine scenario, Alice (P_A) is the client, the user interface (UI_A) is the ATM's set, and p_A is the software executing the client instruction that connects the ATM with the bank (p_B). The context (C_A) is where the ATM is located, a street or the interior of a bank's hall. In a scenario where Alice is accessing a protected web page, the web interface is (UI_A), the browser is the process p_A that runs a protocol with the web server hosting the page, which is process (p_B). The context C_A can be Alice in her office, or in an airport's hall. In a scenario with a few persons collaboratively editing a file in the cloud, the persons are the Alices and Bobs, their screens and keyboards the human-computer interfaces, the software they use to edit and to browse are the processes. The communication happens via the cloud service. The context can be where those persons are, at work, at home, the latter being not only the location but also social environments.

Attack and defence models. STEAL comes also with an attack and with a defence model. They are both relevant for the security analysis, as security is always evaluated with respect to an attacker with specific capabilities (resp., a defender with specific capabilities). The icons $\triangledown$ (attacks) and $\ominus$ (defences)

indicate where the model assumes attacks can strike and where defences can act.

Whatever the nature of the channels and the messages they carry, an attacker can intercept, modify and inject messages in any of those channels. These are typical abilities ascribed to a Dolev-Yao intruder [22]. However, differently from the classical Dolev-Yao, in STEAL, the attacker controls not only the network but also the interactions between the application, the user's interfaces, the persona, and the context. Therefore, an attack may be technical and or a social engineering kind of attack.

Defences also act by interfering with the communication channels. This includes the channels with the user. In our framework, users can participate to improve security, a substantial difference between our and other works [13].

Other assumptions. Our reference model assume that the observable behaviour of the system's elements under analysis is (at least at the level of abstraction chosen) known. However, it does not assume, and does not depend on, the reasons, or the logic, behind this behaving be necessarily understood. This assumption endorses a computational approach. A component (whatever it is, human, interface, agent or context) is an entity (an automaton) that behaves according to a certain control logic that determines its input, output and internal *actions* depending on its *state* and on its (previous) inputs.

For example, a user at an ATM machine, behaves according to some beliefs, desires and intentions that he/she has (withdraw money) which, according to his/her state of mind (I have inserted a pin and wait for the money to come out), determine the actions he/she does (taking the money once out). In its turn the ATM machine's logic is its software code, its state is the machine's state (pin inserted, now checking it), and its actions (display selection of banknotes).

In practice, we may not be able to define precisely a component's control logic, or to list the full set of actions it can ever perform, or to know the component's state in time. But, to build a sufficiently consistent picture (i.e., model) of the component's observable behaviour, one can apply indirect methods to inquire properties about an element's state and to test propositions about it, or by observing the actions it does. For example, we can build a model of a browser by looking at its code. In this case we know fully how it works. If the code is proprietary, we may not be able to fully know its logic but we can build a consistent model by walking through its behaviour. Similarly, we can observe a user interacting with our browser, but we may not be able to observe him changing his mental state (e.g., cognitive process), nor knowing why users behave in certain ways. We can only observe and ask him (e.g., questionnaire/interview).

This assumption is also motivated by the tools of analysis we are going to have: tools for a formal analysis such as model checkers, for the technical security, and human computer interactions methodologies, as those used in usability laboratories, for the social security.

3.2 Methodologies for socio-technical security analysis

STEAL has two methodologies for security analysis. One is apt to understand the security properties without considering a complex model of user's behaviour. The other is apt to question hypotheses on human behaviour and on security properties with the human in the loop.

The two methodologies, together, make the socio-technical analysis possible. The technical analysis helps, against specific threats, discovering if attacks are possible. However, their effectiveness may depend on some user's decisions, exactly as it happens with TLS authentication, where a user may decide to proceed despite a warning flashing that the certificate is invalid. The experimental analysis answers whether those attack would be successful with real users and factual behavioural patterns. The outcomes of the social-oriented analysis also enlighten us on what factors influence critical decisions that may lead to attacks. Such outcomes may therefore suggest defences which, in turn, can be implemented at a technical or a social level or as a combination of them, and understanding their effectiveness triggers another round of analysis. Moreover, it is also possible to perform a security analysis against attacks purely against the human, like social engineering. At the current status of research there is not a stable theory able to model such attacks in a formal model way, thus to study their effect is again done experimentally. This can change in the near future.

To test hypothesis of user's behaviour under socio attacks, we may need to launch such those attacks and harvest the data for analysis. This requires an authorization from an ethical committee and a compliance with a legal framework, assurances that strictly must comply with ethical requirements (APA). In certain situations this may be hard to achieve.

Technical focus - this methodology helps discovering whether an attack is present, within the defined threat model, and mostly with technical interactions and a simple user model. The technical security analysis is applied to elements from UI_A till p_A and possibly p_B till UI_B, including the context(s). P_A is modelled as a non-deterministic process i.e., interacting with process UI_A in every possible way [23, 24, 9]. The technical analysis, can use formal tools of protocol analysis (e.g., model checking [25]), with the only difference that communications are now multi-layered. In a simple case, the analysis can be pursued informally.

Analysing security in this focus means to verify whether specific security properties remain valid despite an intruder. The technical analysis may reveal vulnerabilities due to a faulty integration between the technical and the human layers, like it happens when a system does not offer users to change a password, when it should (e.g., [21]). The output of the technical analysis gives ground for a successive security analysis with social-focus, as it provides information about what attacks should be considered there.

Social focus - this methodology helps discovering security failures in the human interactions, when a predefined threat model is present, or in presence of specific attacks revealed by the technical analysis. The social analysis focuses on

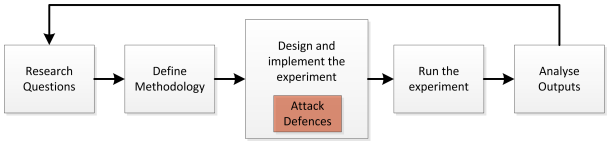

Fig. 2. Social focus: the hypothetico-deductive research model.

human behaviour and choices, therefore from elements S_A till UI_A and possibly their human-to-human interaction with S_B via UI_B, including the context(s). The social analysis uses the hypothetico-deductive model from empirical social sciences research [26] (Fig. 2).

Briefly, the process starts with the initial definition of *research questions* to be tested. These usually come from previous literature review, insights either observed or hinted by human computer interactions. In STEAL they should come from the technical security analysis itself. The process continues with the definition of the most appropriate *research methodology/ies* (i.e., laboratory experiments, interviews, surveys) to answer the research question. Here we also decide on the appropriate threat model and the layers that can be impacted in the reference model. This process is similar if we are testing defences. The next step is to *design and implement the selected methodology(ies)* with the goal of making this process reproducible over a series of experimental tests. After all is set and ready to start, the *experiment is run* and *output data is collected and further analysed*. Usually, data can be analysed using both quantitative (statistical tools can be used to analyse data and test previous defined research questions, and show how significant these are) and qualitative methods (qualitative data gathered from the participants can be correlated with results obtained from statistical analysis and also provide insight or explanation on user's behaviour).

4 Running example: applying STEAL

We describe how STEAL works with a scenario of a visitor at the Univ. of Luxembourg trying to get WiFi Internet access by choosing an *open* SSID name from the list he is presented by his device's network manager.

4.1 Reference model

STEAL reference model highlights the elements of the scenario (Fig. 3), comprising the network manager and all the network communication protocols (p_A), the interface on the user's mobile device (UI_A) and the user trying to select a wireless network name to connect to the Internet (P_A). The premises of the University of Luxembourg, the place where all is happening, is the context C_A.

About the interactions, *express* would be the expression of all the human traits of a persona into how P_A takes security decisions when interacting with

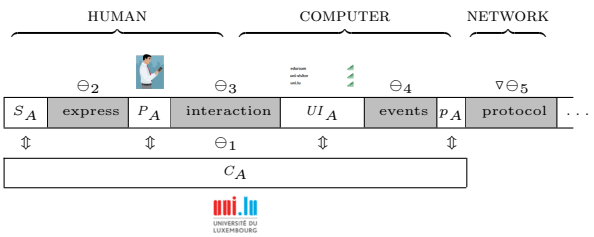

Fig. 3. Reference model for WiFi connection to the Internet.

a human-computer interface in that particular scenario. (We are not able to model those expressions, but we may want to consider them in the analysis). Then, *interaction* are the actions performed by the user, to access the wireless network manager's list and select an SSID name to connect; *events* are the communications exchanged between the user's interface on the mobile device and the wireless network manager application and the wireless access point, which manages calls to its network; *protocol* are the network protocols and messages exchanged between the wireless network manager application and wireless access point, which manages all accesses to the services that its network provides.

4.2 Socio-technical security analysis

Technical analysis - We model the technical layers in a UML diagram. It illustrates the sequence of actions between those elements during an attack in this scenario (Fig. 4). In theory is possible to run a formal analysis against a Dolev-Yao attacker. Here, it is immediately evident that an intruder can open a rogue wireless access point because the SSID is not authenticated.

The success of the attack relies only on the user's choice, precisely on whether a user will actually choose the rogue access point or not. This cannot be understood with this technical analysis only. However, we elaborate more on the attack before passing to the social analysis. We hypothesise that the context plays a very important role in this scenario as the attacker can use the University's visual identity –and all that is connected with it such as knowledge, reputation, etc– to lure a victim to choose a rogue but meaningful name, such as "uni.lu", over the University's official SSID names (actually "uni-visitor" and "eduroam"). The attacker can also set up a second SSID, "secure_AP", a name recalling "security" and test which name has more appeal for the user. Fig. 4 shows the attack.

Social analysis - to apply the hypothetico-deductive model for this analysis we devised the following stages (more detail in [10]): (1) *Research question*: do context and trust influence users' choice of a wireless network name? Alias do names reminding security influence that choice? (2) *Methodology*: on-line survey with two different groups of questions (one relating to context and the other to trust) each together with open questions to provide further explanation of

the participant's selection. The groups of questions must be answered by two different groups of participants (in a between subjects design) regarding wireless network names preferences and graded using a Likert scale (1 - less trusted/less preferred to 5 - highly trusted/preferred); (3) *Design and implement the experiment*: the survey included a list of 12 wireless network names is compiled based on: they exist in the region where the study was conducted, non-existing, evocative of security or freeness and location/context-specific. The participants should be randomly associated with either the first or the second group of questions (between subjects design); (4) *Run the experiment*: send an email to the staff of the University of Luxembourg; (5) *Analysis*: Data was collected, then analysed using R statistical tool. Basic descriptive statistics were applied followed by t-test and wilcoxon rank test. We actually run such an experiment in [10]).

Main results: The social analysis confirms the hypothesis that SSID names reminding the context influence choices, but when users are unaware, or have not been instructed to use the official SSIDs. However, the study refutes the hypothesis that users trust names recalling "security".

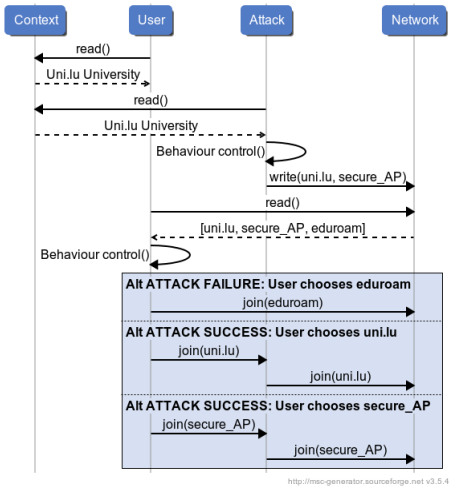

Fig. 4. Technical Focus: the UML sequence diagram for the WiFi connection to the Internet with an intruder attacking the network.

4.3 Adding defences

After having identified possible attacks, we may devise possible defences. We sketch some of them in the reminder of this paragraph. Defences can act at the technical layers or at the social layers. For example, if all wireless access points were strongly authenticated by the user's device, then the identified attack would not occur. This is likely the case with the new Hotspot 2.0, where

the device's SIM card embeds the certificate of proprietary access points. The network manager, p_A, can be programmed to disable the 'join()' action on all networks that have not been vouched by the university's system administrator ($\ominus_4$, Fig. 3). Another technical defence can be implemented at the Network layer by monitoring the live SSIDs, and spot whether some new SSID is trying to use the name of the context (e.g., the "uni.lu" SSID). Technically it is possible to disrupt the joining process to newcomer SSID by sending spoofed deauth packets. This action has the effect of disabling the 'join()' function ($\ominus_5$).

If no technical solution is feasible, defences can be applied at the social layers or to the context. For example, stickers can be left all over the University campus, advertising the legitimate access point of the University ($\ominus_1$). This may likely increase user's awareness. The University can give training to its employees to help them recognize rogue SSIDs ($\ominus_2$). The network manager and the user interface can have a trust indicator displayed aside each SSID ($\ominus_3$).

Whether these defences are effective in successfully decreasing the number of people that fall victim of the attack herein described, is a research question that should be tested by new runs of our framework.

5 Discussion

Conflicts between security and usability are a well known problem. What this paper intends to highlight is that effective security should be the result of a multidisciplinary research. Computer scientists and social scientists must collaborate on similar ground and terminology to study a system's security in an integrated fashion. Security analysis must comprise technical, social and contextual elements. Although the literature has plenty of interesting studies on usability and security, we miss a common operational framework to systematically perform an analysis of security tackling both technical and social aspects.

STEAL comprises a model of a socio-technical scenario and suggests methodologies to analyse and test the same scenario for its security. It helps modelling socio-technical attack scenarios too. At the moment, the methodologies for both technical and social security analysis are working in a pipeline, and allows more runs of analysis. The technical analysis justifies the presence of technical attacks, and the social analysis give ground to evaluate the effectiveness when user's decisions are in place with those attacks. The technical analysis cannot, at the moment, help with attacks of purely social nature, because there is no model able to express and simulate them. The same relates to mature human behaviour: there are no stable human behavioural models that can be used within an automatic security validation tool. Defining such a model must, however, be supported by experimental research.

It is not a primary goal of STEAL to build a model for understanding why users behave the way they do. However, it is possible to use STEAL to design and perform experiments that focus on understanding why some users fall victims of a specific socio-technical attack, by following some behavioural patterns. Such findings may inspire defences, whose effectiveness can be tested in STEAL.

We showed here how to apply our framework in a specific socio-technical scenario. However, we need more examples to be more confident about the flexibility of the approach. Regarding our reference model, this has been shown as we applied it to model socio-technical scenario about users accessing the Internet [9, 10], but more scenarios are needed to validate the actual flexibility. About the technical security analysis, it can be applied generically once all components of a socio-technical system, together with its interactions, are modelled as suggested. The main issue to consider is the human behavioural analysis, and if STEAL can help to generalize this analysis for a large set of scenarios. We believe that the methodology used for the social analysis (hypothetico-deductive experimental model) is generic enough to be applied in the design and implementation of user related experiments for socio-technical systems. In order to perform the security analysis, all the steps of that process need to be clearly and objectively defined. It may be the case that we can only test one interaction, and therefore, one hypothesis at a time. Still, its analysis uses methods that either confirm or dismiss that hypothesis. Once we know this answer we can step to the next question or generate some conclusion. This is still generic and prone to be adapted to different socio-technical scenarios. As discussed in the methodology, some experiment may need authorization from an ethical committee, compliance with a legal framework, and with ethical requirements (APA), before being set.

6 Conclusion

We believe that STEAL is a good first step in the integration of socio-technical security analysis by a multidisciplinary team. Nevertheless, there is the need to apply STEAL to model and analyse more socio-technical scenarios. Only this way will it be possible to improve STEAL and enrich its flexibility and generalization. As future work we plan to use STEAL to design and test the devised socio-technical defences for each scenario and verify whether they work or need further revision.

References

1. R. West, "The Psychology of Security," *Communication of the ACM*, vol. 51, no. 4, pp. 34–38, April 2008.
2. A. Tversky and D. Kahneman, "Judgment under uncertainty: Heuristics and biases," *Science*, vol. 185, pp. 1124–1131, 1974.
3. P. Kumaraguru, S. Sheng, A. Acquisti, L. F. Cranor, and J. Hong, "Teaching johnny not to fall for phish," *ACM Trans. Internet Technol.*, vol. 10, no. 2, pp. 7:1–7:31, Jun. 2010.
4. S. Parkin, A. van Moorsel, P. G. Inglesant, and M. A. Sasse, "A Stealth Approach to Usable Security: Helping IT Security Managers to Identify Workable Security Solutions," in *Proc. of NSPW 2010, Sept. 21-23, 2010.* ACM, 2010, pp. 33–50.
5. R. Tembe, K. W. Hong, E. Murphy-Hill, C. Mayhorn, and C. Kelley, "American and indian conceptualizations of phishing," in *Proc. of STAST 2013.* IEEE, pp. 37–45.

6. M. Volkamer, S. Stockhardt, S. Bartsch, and M. Kauer, "Adopting the cmu/apwg anti-phishing landing page idea for germany," in *Proc. of STAST 2013*. IEEE, pp. 46–52.
7. J. E. Bardram, "The trouble with login: on usability and computer security in ubiquitous computing," in *Proc. of Personal and Ubiquit. Comput.*, ser. LNCS, vol. 9, no. 6. Springer, 2005, pp. 357–367.
8. D. Weerasinghe, V. Rakocevic, and M. Rajarajan, "Security framework for mobile banking," in *Proc. of the 8th MoMM 2010*. ACM, 2010, pp. 421–424.
9. A. Ferreira, R. Giustolisi, J. Huynen, V. Koenig, and G. Lenzini, "Studies in socio-technical security analysis: Authentication of identities with tls certificates," in *Proc. of the 12th IEEE TrustComm-13*, 2013.
10. A. Ferreira, J. Huynen, V. Koenig, G. Lenzini, and S. Rivas, "Socio-technical study on the effect of trust and context when choosing wifi names," in *Proc. of the 9th ESORICS 2013*, 2013.
11. F. Zhu, S. Carpenter, A. Kulkarni, and S. Kolimi, "Reciprocity attacks," in *Proc. of the SOUPS 2011*. New York, NY, USA: ACM, 2011, pp. 9:1–9:14.
12. I. Arce, "The weakest Link Revisited," *Security Privacy, IEEE*, vol. 1, no. 2, pp. 72 – 76, mar-apr 2003.
13. L. F. Cranor, "A Framework for Reasoning About the Human in the Loop," in *Proc. of the 1st Conf. on Usability, Psychology, and Security*. USENIX Association, 2008, pp. 1–15.
14. G. Conti, M. Ahamad, and J. Stasko, "Attacking information visualization system usability overloading and deceiving the human," in *Proc. of the SOUPS 2005*. ACM, 2005, pp. 89–100.
15. L. Falk, A. Prakash, and K. Borders, "Analyzing websites for user-visible security design flaws," in *Proceedings of SOUPS 2008*. New York, NY, USA: ACM, 2008, pp. 117–126.
16. A. De Luca, M. Langheinrich, and H. Hussmann, "Towards understanding atm security: a field study of real world atm use," in *Proc. of SOUPS 2010*. New York, NY, USA: ACM, 2010, pp. 16:1–16:10.
17. L. Janczewski and F. Lingyan, "Social engineering-based attacks: Model and new zealand perspective," in *Proc. of IMCSIT 2010*, 2010, pp. 847–853.
18. F. Dalpiaz, P. Giorgini, and J. Mylopoulos, "Adaptive Socio-Technical Systems: a Requirements-driven Approach," *Requirements Engineering*, pp. 1–24, 2013.
19. K. Worton, "Using socio-technical and resilience frameworks to anticipate threat," in *Proc. of STAST 2012*, 2012, pp. 19–26.
20. N. Pavkovic and L. Perkov, "Social engineering toolkit x2014; a systematic approach to social engineering," in *Proc. of MIPRO 2011*, 2011, pp. 1485–1489.
21. G. Bella and L. Coles-Kemp, *Layered Analysis of Security Ceremonies*. Springer, 2012, vol. 376, pp. 273–286.
22. D. Dolev and A. Yao, "On the security of public-key protocols," *IEEE Transaction on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.
23. G. Bella, R. Giustolisi, and G. Lenzini, "Socio-Technical Formal Analysis of TLS Certificate Validation in Modern Browsers," in *Proc. of PST 2013*. IFIP, 2013.
24. ——, "A Socio-Technical Understanding of TLS Cerficate Validation," in *Proc. of 7th IFIPTM2013*. IFIP, 2013.
25. E. M. Clarke, O. Grumberg, and D. Peled, *Model Checking*. MIT press, 1999.
26. P. Godfrey-Smith, *Theory and Reality: An Introduction to the Philosophy of Science*, ser. Science and Its Conceptual Foundations. Univ. of Chicago Press, 2009.